REMARKS

This paper amends Claims 21, 25 and 32, and cancels Claims 29 and 31. Claims 1-11, 13-20, 22-24, 26-28 and 33-37 are unchanged. Claims 1-11, 13-28 and 32-37 are pending. Reconsideration and allowance of the claims in light of the present remarks is respectfully requested.

A clerical error has been corrected in Claim 21 by adding the missing preposition "to". This amendment is not done to narrow the claim or to avoid any cited references.

Discussion of Rejections under 35 U.S.C. § 102(b) and § 103(a)

Claims 1-10, 25-28 and 32-34 were rejected under 35 U.S.C. § 102(b) as being anticipated by Singh (Singh, K.J., "Performance Optimization of Digital Circuits, Ph.D. Dissertation, University of California Berkeley, 1992), referred to as *Singh '92*. Claims 21-24 were rejected under 35 U.S.C. § 102(b) as being anticipated by Singh et al. (Singh et al., "Timing Optimization of Combinational Logic," ICCAD-88, Digest of Technical Papers, IEEE International Conference, 7-10 Nov. 1988, pages 282-285), referred to as *Singh '88*.

Claims 29 and 31 were rejected under 35 U.S.C. § 102(b) as being anticipated by *Huang* et al., ("An Iterative Area/Performance Trade-Off Algorithm for LUT-Based FPGA Technology Mapping," Computer-Aided Design, 1996, ICCAD-96, Digest of Technical Papers, 1996 IEEE/ACM International Conference on 10-14 Nov. 1996, pages 13-17).

Claims 35-37 were rejected under 35 U.S.C. § 103(a) as being obvious over Singh '92 as applied to Claim 1 above, and further in view of Singh et al. ("A Heuristic Algorithm for the Fanout Problem," 27th ACM/IEEE Design Automation Conference, 1990), referred to as *Singh '90*.

Claims 11-20 were rejected under 35 U.S.C. § 103(a) as being obvious over *Ju* et al. ("Incremental Techniques for the Identification of Statistically Sensitizable Critical Paths," 28th ACM/IEEE Design Automation Conference, Paper 32.2, pages 541-546, 1991), and further in view of Singh '92.

Regarding independent **Claim 1**, the identification of O_1 as the most critical output in Singh '92 only identifies the most critical output for the purpose of judging if a circuit meets the timing specification or not. Identification of the most critical output in Singh is not used to order a node on which to do a transformation whatsoever.

Applicant respectfully submits that the citation of section 3.2.1, second sentence and the citation on page 57

> ...in a tree, the slack is non-decreasing along any path from input to output. This implies that for node y that is a fanin of node x, $s(x) \leq s(y)$

of Singh '92 does not describe "**sorting fanins of the first node** according to slack values associated with the corresponding fanins, wherein at least a portion of the slack values differ in value." The second sentence of section 3.2.1 has nothing to do with ordering the fanins of the first node, but instead describes **ordering primary outputs**, which is done to identify the most critical output for checking purposes. The citation on page 57 is only a statement of fact between a node and one fanin of the node in terms of their slack relationship. However, this does not disclose sorting fanins of the first node according to slack values associated with the corresponding fanins, where at least a portion of the slack values differ in value, as recited by Applicant.

Applicant respectfully submits that the citation of Heuristic-2, sentences 1-6 and Heuristic-3, sentences 1-2 on page 60 of Singh '92 does not disclose "reducing delays associated with fanins having relatively larger negative slack values before reducing delays associated with fanins having relatively smaller negative slack values." The citations on page 60 describe reducing the delay of the most critical primary output of a circuit first. Singh '92 does not order the fanins of that primary output, and therefore, does not reduce delays associated with fanins having relatively larger negative slack values before reducing delays associated with fanins having relatively smaller negative slack values.

Applicant respectfully submits that the citation of Singh '92, Figure 3.5 on page 45 and page 60, Heuristic-1 "several iterations may be required to meet the achievable target" does not disclose "wherein reducing delays is performed recursively" as recited in dependent **Claim 2**. The figure shows pseudo-code that has three steps and an end condition. There is nothing recursive in the cited pseudo-code. Furthermore, performing iterations as done by Singh is not the same as performing the reducing recursively.

Regarding independent **Claim 6**, the above discussions with respect to Claims 1 and 2 apply here as well. Further the cited Heuristic-2 and Heuristic-3 of Singh '92 describe the transformations statically. In contrast Applicant recites "recursively reducing delays associated with at least a portion of the critical fanins of the first node", which is a dynamic act. Furthermore, Singh does not disclose a recursive act as claimed by Applicant.

Regarding independent **Claim 9**, the cited text of Singh '92 identifies the possible candidates for transformation (the ε-critical netlist) and another algorithm is used to determine the transformations. In contrast, Applicant claims a recursive algorithm to reduce delay where the delay reduction at transitive fanins of a node reduces the delay of the current node. The two methods are very different.

The Office Action applies the Ju reference with respect to independent **Claim 11**. The disclosure of Ju is in a completely different domain from Claim 11. Ju describes incremental timing analysis and simulation, and how to identify false paths and efficiently enumerate critical paths so as to update timing more efficiently. The Ju reference does not disclose timing optimization as claimed by Applicant. For example, the Ju reference does not disclose "A method of reducing timing delays" as recited by Claim 11.

The Office Action applies the Singh '88 reference with respect to independent **Claim 21**. Singh '92 is Dr. Singh's Ph.D. dissertation and it covers and improves the work done in Singh '88, therefore, the above discussions distinguishing Applicant's claims from Singh '92 apply to Singh '88 as well. The Singh '88 reference discloses using the static method, as opposed to Applicant's dynamic method of reducing delays recited in Claim 21. The static method described in Singh '88 pre-selects a set of nodes for transformation based on the current static timing and it does not consider the impact of performing one node's transformation on the other nodes. Therefore, the transformations as disclosed by Singh are not done accurately based on up-to-date timing and further, some transformations are simply wasted. Claim 21 recites a dynamic method of reducing delays where timing analysis is updated <u>dynamically</u> so that each transformation is performed with up to date and accurate timing on its fanins.

Claim 21 recites in part: "reducing a first critical path delay at a first node in closer proximity to a primary input associated with the critical path than to a node in closer proximity the primary output; storing the reduced delay". The Office Action identified section 2.5, last paragraph of Singh '88, as disclosing these acts. The Singh '88 reference describes that, at each iteration, a set of nodes is picked for transformation. After those transformations are done, the reference describes updating timing for the entire design so that the timing is correct when the next set of nodes is picked for decomposition. However, in one iteration, these transformations can impact the delays of the other transformations, i.e., when one transformation is done, the arrival times are not updated based on other transformations in the same iteration.

Claim 21 further recites in part: "recursively reducing a second critical path delay beginning at a second node located between the first node and the primary output based at least in part on the stored reduced delay." The Office Action identified Figure 2 as disclosing this act. Applicant respectfully submits that the speedup_node code is not a recursive algorithm that can be used to dynamically adjust the updated delay. There is no recursiveness shown in Figure 2. The top level function is speed_up(network, dist, e), where the sub-function it is calling is speedup_node(node). There is no recursiveness there. In Figure 4, the speed up for a particular node (a collapsed node) is done in a recursive manner, but that is only for this particular node, and does not involve any other part of the netlist.

Regarding independent **Claim 25**, the Singh '92 reference describes that all outputs of the circuit have the same delay target. In contrast, Applicant's amended Claim 25 permits different delay targets at different outputs, which is advantageous for real designs and applications. Furthermore, Singh '92 uniformly reduces the circuit delay along iterations, while Applicant claims dynamically determining the amount of delay to reduce based on the history of previous delay reduction iterations. This feature can provide better results in comparison to the results of Singh '92.

Regarding dependent **Claim 32**, Applicant respectfully submits that the cited text of Singh '92, section 3.2.1, paragraphs 1 and 2, does not disclose "performing a local transformation on the first node if the reducing delays for at least one of a set of fanins with negative slack values of the first node is not successful".

Regarding dependent **Claim 33**, Applicant respectfully submits that the cited text of Singh '92, section 3.2.1, first paragraph, does not disclose "wherein reducing delays associated with the fanins of the first node is performed before any local transformation of the first node." The Office Action identifies "specifically the last two lines, wherein a local transformation is needed since the circuit performance was unchanged" as describing the claimed feature. This is opposite to the claim where reducing delays associated with the fanins of the first node is performed before any local transformation.

Regarding dependent **Claim 34**, Applicant respectfully submits that the cited text of Singh '92, section 3.2.1, second sentence and the phrase on page 57, do not describe "wherein sorting fanins of the first node includes sorting fanins of the first node in order according to slack values associated with the corresponding fanins" as claimed. The second sentence of section 3.2.1 has nothing to do with, and is not concerned with ordering the fanins of the first node, but instead describes **ordering primary outputs**, which is done to identify the most critical output for checking purposes. The citation on page 57 is only a statement of fact between a node and one fanin of the node in terms of their slack relationship and does not describe the fanin sorting feature.

The Office Action applies the Singh '92 and the Singh '90 references regarding dependent **Claim 35**. Singh '92 is Dr. Singh's Ph.D. dissertation and it covers and improves the work done in Singh '90, therefore, the above arguments distinguishing Applicant's claims from Singh '92 apply to Singh '90 as well. Furthermore, performing *iterations* as disclosed by Singh is not the same as performing the delay reduction *recursively* as recited by Claim 35. Iteratively performing the delay reduction and recursively performing the delay reduction are simply not the same.

The Office Action applies the Singh '92 and the Singh '90 references with respect to dependent **Claim 36**. The Singh references describe a static method as opposed to Applicant's dynamic method, such as explained in the discussion of Claim 21 above. The Singh references describe that, in each iteration, a few nodes are picked for transformation without knowing the impact that each transformation had on the other transformations.

The steps of the buffering algorithm in Singh are repeated for the set of nodes in a "while loop" as long as the delay decreases. Claim 36 recites: "wherein reducing delays associated with the fanins of the first node further comprises stopping after c) if the delay reduction of one of the fanins is not successful". This is in contrast to what Singh does regarding repeating in the "while loop". The "while loop" in Singh's papers is iteration, which is in contrast to the recursive technique claimed by Applicant. A fundamental difference between the two techniques is what is done inside the while loop. Singh describes making static calculations to get a set of nodes and then doing transformations. Applicant's approach does not compute a set of nodes upfront, but rather computes and transforms them on the fly, with updated timing out of each transformation.

Moreover, the Singh '90 reference discusses a fanout optimization problem, i.e., buffering, which is a totally different problem than Applicant solves and claims. Furthermore, the basic technique used in '90 reference is essentially the same as in the Singh '88 reference, which is to, in one iteration, pick a critical node set and perform transformations, which, in this case, is to buffer the fanout net.

Therefore, Applicant respectfully requests the withdrawal of all claim rejections and prompt allowance of Claims 1-11, 13-28 and 32-37.

Conclusion

In light of the above, reconsideration and withdrawal of the outstanding rejections are specifically requested. In view of the foregoing remarks, Applicant respectfully submits that the claims of the above-identified application are in condition for allowance. However, if the Examiner finds any impediment to allowing all claims that can be resolved by telephone, the Examiner is respectfully requested to call the undersigned.

Please charge any additional fees, including any fees for additional extension of time, or credit overpayment to Deposit Account No. 11-1410.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: August 9, 2005

By:
Raimond J. Salenieks
Registration No. 37,924
Agent of Record
Customer No. 20,995
(619) 235-8550